IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

DO — Atty. Ref.: 2687-8

Serial No. 09/412,334 — Group: 2126

Filed: October 5, 1999 — Examiner: Anya, Charles

For: ARRANGEMENT FOR SIMPLIFYING THE DESIGN AND IMPLEMENTATION OF MOBILE SERVICES IN A COMMUNICATION SYSTEM

Before the Board of Patent Appeals and Interferences

BRIEF FOR APPELLANT

On Appeal From Final Rejection
From Group Art Unit 2126

John R. Lastova
NIXON & VANDERHYE P.C.
8th Floor, 1100 North Glebe Road
Arlington, Virginia 22201-4714
(703) 816-4025
Attorney for Appellant
DO
Telefonaktiebolaget LM Ericsson (publ)

TABLE OF CONTENTS

TABLE OF AUTHORITIES

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

DO	Atty. Ref.: 2687-8
Serial No. 09/412,334	Group: 2126
Filed: October 5, 1999	Examiner: Anya, Charles

For: ARRANGEMENT FOR SIMPLIFYING THE DESIGN AND IMPLEMENTATION OF MOBILE SERVICES IN A COMMUNICATION SYSTEM

* *

January 3, 2005

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Telefonaktiebolaget LM Ericsson (publ), a Swedish corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals related to this subject application. There are no interferences related to this subject application.

III. STATUS OF CLAIMS

Claims 52-76 are pending. Claims 52-55, 57, 59, 60, and 68-76 stand rejected under 35 U.S.C. §103 as being unpatentable over "Engineering Modelling Concepts (DPE Architecture) Version 2.0" to Tina-C Deliverable, (hereafter "Tina-C"), in view of U.S. Patent 5,517,677 to Moon. Claims 56 and 65 stand rejected under 35 U.S.C. §103 as being unpatentable over Tina-C, Moon, and Puder. Claims 61-64 stand rejected under 35 U.S.C. §103 as being unpatentable over Tina-C, Moon, and Tokmakoff. Claims 58, 66, and 67 stand rejected under 35 U.S.C. §103 as being unpatentable over Tina-C, Moon, and Leydekkers.

IV. STATUS OF AMENDMENTS

It is uncertain whether the amendment filed after final on September 7, 2004 is entered for purposes of Appeal. To date, Applicant has not received any communication from the Examiner acknowledging or responding to the after final amendment. In telephone conversations with the Examiner in early October 2004, he indicated that the file was inaccessible because it was being imaged.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The claims are directed to an arrangement for simplifying the design and implementation of mobile services in a distributed communications system. A distributed communications system employs distributed hardware and software. In Open Distributed Processing (ODP) adopted by the Telecommunication Information Networking Architecture (TINA), a telecommunication application is realized by a set of interacting computation objects which rely on an abstract infrastructure called a distributed processing environment (DPE). Figure 1 illustrates a DPE example. A DPE node is a unit of resource administration that provides support to the DPE architecture. The DPE "hides" the complex details of mechanisms used to overcome problems associated with distribution and is commonly referred to as "distribution transparency" in the ODP reference model.

The distribution transparenices recognized and defined by ODP/TINA are listed on page 2 of the specification. But these defined distribution transparencies do not support mobility. Mobility allows the communications system to monitor a current location of a mobile user equipment and to support a connection with a mobile user equipment, even as the mobile user equipment moves across various geographic coverage areas. Mobility is offered in existing cellular communications systems like GSM, UPT, and others but not in a *transparent* way.

Because ODP/TINA provides an efficient framework for designing and implementing distributed applications in communications networks, it was assumed that mobility would be inherently supported. That assumption was wrong. In order to design a new communications application, the designer must therefore be knowledgeable about and specifically design in mobility support functions. But this defeats the point of distribution transparency.

The invention adds mobility transparency to the distribution transparencies that exist in the ODP adopted by TINA. Mobility transparency hides the various mechanisms and functions needed to support mobility for any new mobile communications application being designed. As a result, such mobile communications applications may be designed more simply and efficiently.

The detailed description offers three non-limiting, example alternatives for adding mobility transparency to a distributed processing based communications system referred to as: the "in-line alternative," the "broker alternative," and the "proxy alternative." Each alternative introduces necessary mobility functions M which include, for example, terminal support, personal support, session support, and mobility-related support. By providing mobility transparency, the design and implementation of mobile applications are greatly simplified and are no more complicated than the design and implementation of fixed applications. This also allows existing fixed applications to be transformed into mobile applications.

Claim 52 recites "means for supporting one or more transparencies in the DPE including access, location, or failure transparencies." Reference is made to the DPE physical infrastructure shown in Figure 1 and to the access, location and failure distribution transparencies described on page 2 implemented using hardware and software in the DPE infrastructure. Claim 52 recites "means for supporting mobile radio terminal mobility transparency in the DPE." Reference is made to the DPE physical infrastructure shown in Figure 1, to the description of the mobility functions introduced described on pages 7-8, and to Figures 3, 5-7, 11, and 12 along with the supporting text related to supporting mobility transparency in the DPE implemented using hardware and software in the DPE infrastructure.

Claim 55 includes "means for mapping computational objects to engineering objects (EO) so as to be non-visible in a computational model of the application program." Reference is made to Figures 3, 6, and 12 which illustrate mapping examples implemented using hardware and software in the DPE infrastructure.

Claim 57 includes "means for effecting an interaction between computational objects belonging to a same cluster is effected directly using one or more method calls," and "means for effecting communication between computational objects located in a telecom system domain and in different clusters through a channel including stubs, binders, and protocols." Reference is made to

Figure 3, 6, 8, and 12 and the associated descriptive text which are implemented using hardware and software in the DPE infrastructure.

Claim 60 includes "means for sending a message to a routing broker." Reference is made to Figure 4 and supporting text implemented using hardware and software in the DPE infrastructure.

Claim 74 includes "means for registering in the mobility function first and second objects in different domains." Reference is made to Figures 7 and 8 along with supporting text implemented using hardware and software in the DPE infrastructure. Claim 74 also includes "means for defining and registering a proxy for the object the proxy represents." Reference is made to Figures 9-12 along with supporting text implemented using hardware and software in the DPE infrastructure.

Claim 76 includes "means for introducing a redirection function on the DPE and for generating a special stub for a dynamic object." Reference is made to Figure 11 and supporting text implemented using hardware and software in the DPE infrastructure.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The primary rejection to be reviewed on appeal is the obviousness rejection based on Tina-C and Moon. In addition, the rejections based on Tina-C, Moon,

and Puder; Tina-C, Moon, and Leydekkers; Tina-C, Moon, and Tokmakoff should also be reviewed.

VII. ARGUMENT

A. Tina-C and Moon Do Not Disclose or Suggest Claim 52

The Examiner states Tina-C "teaches an arrangement for simplifying the design and implementation of mobile services in a communications system," referring to pages 5-9/5-10. Applicant respectfully disagrees. Tina-C does not teach simplifying the design and implementation of mobile communication services. The text in lines 5-9 and 5-10 relates to service and user session managers and describes processing a new video conference service session request (see Figure 5-5). There is no teaching here of mobile services.

The Examiner then goes on to admit that Tina-C fails to disclose:

> means for supporting mobile radio terminal mobility transparency such that an application program being executed at a mobile ratio terminal located in one radio service area serviced via one radio base station is not interpreted or hindered in its execution when the mobile radio terminal moves to another radio service area serviced via another radio base station.

For these deficiencies, the Examiner turns to the Moon patent which relates to mobile radios scanning various channels in a trunked radio system to determine if an incoming call is directed to that radio. See column 1, lines 42-45.

The problem to which Moon is directed is the large number of home channel scannings that a trunked mobile radio must perform in multi-system areas. The mobile radio must scan the home channel of each authorized system and compare various identification codes to determine if an incoming call is intended for the mobile radio. See column 2, lines 15-23 as well as column 3, lines 24-31. Moon uses an adaptive queue embedded in the scanning sequence in which entries are continually updated based upon a metric that is shown by past history to be more often used by this mobile radio than other sequence entries. See column 3, lines 51-57. This adaptive queue is particularly helpful when the mobile radio is roaming between different geographical areas, as is explained in column 8, lines 29-38, referenced by the Examiner.

Mobility has been supported for roaming mobile terminals in trunked radio systems and in conventional cellular radio systems for quite some time. But claim 52 is not simply directed to supporting mobile radio terminal mobility as it appears the Examiner is suggesting. To the contrary, claim 52 recites supporting mobile terminal:

> *mobility transparency in the DPE* such that an application program being executed at a mobile terminal located in one radio service area serviced via one radio base station is not interrupted or hindered in its execution when the mobile radio terminal moves to another radio service area serviced via another radio base station.

First, there is no teaching of a distributed processing environment in Moon. Second, Moon's adaptive scanning technique, which facilitates mobile terminal roaming, does not describe or support *mobility transparency* in a *distributed processing environment.* As a result, applications designers in Moon's system must be aware of the details of how mobility is implemented in order to design new communications applications that support mobility.

Nor is it clear how a trunked radio, adaptive scanning technique for use in a roaming mobile terminal would be employed in a distributed process environment to support mobility *transparency.* First, mobility transparency is supported in the DPE network and not in a mobile terminal. Moon's scanning is performed in the terminal—not in the network. Second, Moon does not detail a particular roaming scheme for supporting mobility transparency even in a centralized processing environment, which is what Moon's conventional trunked communications system is. Third, the particulars of how mobility is accomplished are not disclosed by Moon. Fourth, the handover and roaming mechanisms that might be used in a mobility scheme in Moon are presumably centralized rather then distributed.

B. There Is No Legal Motivation to Combine Tina-C and Moon

The Examiner suggests that Moon's mobile radio scanning could be combined with Tina-C because it would improve the Tina-C system "by providing enhanced roaming feature of mobile radios and telephone." But Tina-C does not

disclose any roaming feature, and therefore, there is nothing to be enhanced. There also is no teaching in Moon or Tina-C of supporting any kind of *mobility transparency in a distributed processing environment.* Nor does either describe how mobility transparency would be implemented in a distributed data processing environment.

The Federal Circuit prohibits:

> rejecting patents solely by finding prior art corollaries for the claimed elements [because this] would permit an Examiner to use the claimed invention itself as a blueprint for piecing together elements in the prior art to defeat the patentability of the claimed invention.

In re Rouffet, 149 F.3d 1350, 1357 (Fed. Cir. 1998). Such an approach would be "an illogical and inappropriate process by which to determine patentability." *Sensonics, Inc. v. Aerosonic Corp.* 81 F.3d 1566, 1570 (Fed. Cir. 1996). Yet, this is the very approach that the Examiner is taking in an attempt to combine Moon with Tina-C. There is no motivation to combine them absent the motivation of trying to reject the instant claims.

The *Rouffet* Court stated, "the Examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and no knowledge of the claimed invention, would select the elements from the prior art references for the combination in the mannered claimed." *In re Rouffet,* 149 F.3d at 1357. The Examiner has not shown where either Tina-C or Moon recognized or confronted

the same problem as the inventor—namely, how to provide mobility transparency in a distributed processing environment.

Indeed, in November 1997, the inventor delivered a paper at a TINA-C Conference in Chile entitled, "Terminal Mobility Support in TINA-C."[1] That article demonstrated that the TINA-C architecture does not support mobile terminal mobility. See, for example, the Abstract of this article which states "as shown in [1] Access and Location transparencies defined for Open Distributed Processing (ODP) and TINA-C are insufficient to support terminal mobility since interoperability between DPE platforms is not guaranteed at all times" (emphasis added).

Neither Tina-C nor Moon disclose any solution to this inoperability between DPE platforms which would be necessary to support roaming and terminal mobility. Thus, the combination of Tina-C and Moon is not proper as a matter of law. In view of that fact and the fact that their combined teachings fail to disclose the claimed "means for supporting mobile radio terminal mobility transparency in the DPE...," the rejection of independent claim 52 is improper and should be reversed.

[1] A copy of this article is provided in the Evidentiary Appendix.

C. Tina-C and Moon Do Not Disclose or Suggest Claim 55

Applicant has been unable to locate where the Tina-C reference describes mapping computational objects to engineering objects "so as to be non-visible in a computational model of the application program." The burden of proof to show where a claim feature is described in a reference is on the Patent Examiner. *In re Piasecki*, 223 USPQ 785, 788 (Fed. Cir. 1984); 37 CFR 1.104(c)(2).

D. Tina-C and Moon Do Not Disclose or Suggest Claim 57

Regarding claim 57, Applicant is unable to locate in the text referred by the Examiner where the claimed interaction between computational objects is described or where communication is effected between those objects "through a channel including stubs, binders, and protocols." In paragraphs 13 and 14 of the Office Action, the Examiner refers to "Chapman," which is not a reference identified in the formal statement of the final rejection.

E. Tina-C and Moon Do Not Disclose or Suggest Claim 68

Regarding claim 68, page 5-5 of Tina-C describes an object in a group creating another object via interaction with the group manager, which is not the same thing as a proxy object acting on behalf of an entity.

F. <u>Tina-C and Moon Do Not Disclose or Suggest Claim 69</u>

Regarding claim 69, there is no mention on page 5-5 of invoking a mobility function operation.

G. <u>Tina-C and Moon Do Not Disclose or Suggest Claim 70</u>

A proxy object being a symmetrical consultation as recited in claim 70 is not described on page 5-4 of TINA-C. Where in page 5-4 is an object described as being represented by multiple objects?

H. <u>Tina-C and Moon Do Not Disclose or Suggest Claim 76</u>

Applicant has reviewed page 5-5 relied on by the Examiner. This page describes creating an object in a group. There is no teaching of introducing a redirection function on the DPE and generating a special stub for a dynamic object as recited in claim 76.

I. <u>The Other References Do Not Remedy Tina-C's Deficiencies</u>

Puder, Tokmakoff, And Leydekkers do not remedy the deficiencies of Tina-C or Moon with respect to independent claim 52. None of the applied references discloses or suggests:

> supporting mobile radio terminal mobility transparency in the DPE such that an application program being executed at a mobile radio terminal located in one radio service area serviced via one radio base station is not interrupted or hindered in its execution when the mobile radio terminal moves to

another radio service area serviced via another radio base station.

VIII. CONCLUSION

Lacking features of the claims as explained above, the Board should reverse the outstanding rejections.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: ______________________
John R. Lastova
Reg. No. 33,149

JRL/kmm
Enclosures
Appendix A - Claims on Appeal

IX. CLAIMS APPENDIX

52. An arrangement for simplifying the design and implementation of mobile radio terminal services in a communications system, comprising:

distributed hardware and software components provided in accordance with a distributed processing environment (DPE) and configurable in use to provide one or more services to one or more users;

means for supporting one or more distribution transparencies in the DPE including access, location, or failure transparencies; and

means for supporting mobile radio terminal mobility transparency in the DPE such that an application program being executed at a mobile radio terminal located in one radio service area serviced via one radio base station is not interrupted or hindered in its execution when the mobile radio terminal moves to another radio service area serviced via another radio base station.

53. Arrangement as claimed in claim 52, wherein the access, location, or failure transparencies are already existing and are defined by Open Distributed Processing (ODP) and adopted by Telecommunication Information Networking Architecture (TINA-C), and wherein the means for supporting mobile radio terminal mobility transparency in the DPE is added to the already-existing the access, location, or failure transparencies.

54. Arrangement as claimed in claim 52, wherein the means for supporting mobile radio terminal mobility transparency in the DPE is introduced at requirement and

functional specification phases by integrating a mobile radio terminal mobility function into an infrastructure of a software platform designed in the DPE.

55. Arrangement as claimed in claim 52, wherein the means for supporting mobile radio terminal mobility transparency in the DPE includes means for mapping computational objects to engineering objects (EO) so as to be non-visible in a computational model of the application program.

56. Arrangement as claimed in claim 52, wherein means for supporting mobile radio terminal mobility transparency in the DPE includes an engineering object interceptor arranged at a boundary between a mobile radio terminal domain and a telecom system domain.

57. Arrangement as claimed in claim 52, wherein an engineering model may be developed by mapping each of one or more computational objects (COs) to one or more Basic Engineering Objects (BEOs), the arrangement further comprising:

means for effecting an interaction between computational objects belonging to a same cluster is effected directly using one or more method calls, and

means for effecting communication between computational objects located in a telecom system domain and in different clusters through a channel including stubs, binders, and protocols.

58. Arrangement as claimed in claim 52, wherein user domain computation objects and a telecom system domain computation object residing in different clusters communicate in a channel including an interceptor transparent to an application designer.

59. Arrangement as claimed in claim 52, wherein an application designer can decide from a computational model whether an object belongs to a user domain or a telecom system domain for use in generating application objects.

60. Arrangement as claimed in claim 52, further comprising:

means for sending a message to a routing broker asking for a server to perform a task,

wherein the routing broker is configured to implement a mobility function to locate the server and send the request to the server.

61. Arrangement as claimed in claim 60, wherein the routing broker is a cascade of two brokers.

62. Arrangement as claimed in claim 61, wherein the two brokers are configured to allow interactions between an object belonging to a user domain and an object belonging to a telecom system domain.

63. Arrangement as claimed in claim 62, wherein the two brokers support both a same interface type containing an invoke operation to allow a request to be built and invoked dynamically by client objects.

64. Arrangement as in claim 63, wherein the invoke operation includes an object name or identifier, an operation name, and a parameter list for the invoked operation.

65. Arrangement as claimed in claim 60, wherein the mobility function is a functional layer in a system architecture between an application layer and a DPE layer, where each layer may use services offered by another layer.

66. Arrangement as claimed in claim 60, wherein a derived computational model is configured for use in a transition from a computational model to an engineering model in the DPE to map interactions traversing a boundary between a user domain and a telecom system domain to interactions with the mobility function.

67. Arrangement as claimed in claim 66, wherein based on the derived computational model, an engineering model can be elaborated and engineering objects can be generated.

68. Arrangement as claimed in claim 60, further comprising:

a proxy object for acting on behalf of an entity in a transparent way.

69. Arrangement as claimed in claim 68, wherein the proxy object is adapted to generate an invoke operation for the mobility function.

70. Arrangement as claimed in claim 69, wherein the proxy object is a symmetrical constellation.

71. Arrangement as claimed in claim 68, wherein each proxy object is a Dynamic Object (DO), and a DO instance may be initiated from an object template corresponding to a Dynamic Object Implementation (DOI).

72. Arrangement as claimed in claim 68, wherein a proxy represents only one object and is deleted when the represented object terminates.

73. Arrangement as claimed in claim 68, wherein an object can be represented by multiple proxies.

74. Arrangement as claimed in claim 68, further comprising:

means for registering in the mobility function first and second objects in different domains, and

means for defining and registering a proxy for the object the proxy represents.

75. Arrangement as claimed in claim 74, wherein objects may be grouped into clusters, capsules, and nodes.

76. Arrangement as claimed in claim 52, further comprising:

means for introducing a redirection function on the DPE and for generating a special stub for a dynamic object.

X. EVIDENCE APPENDIX

Attached is a copy of a November 1997 paper entitled, "Terminal Mobility Support in TINA-C," delivered at a TINA-C Conference in Chile by Do van Thanh and Jan Audestad and a copy of a September 1996 paper entitled, "Mobility and TINA," in the *Proceedings of the TINA 96 Conference*, volume ISBN 0.

XI. RELATED PROCEEDINGS APPENDIX

There is no related proceedings appendix.

Terminal mobility support in TINA

Do van Thanh[a] and **Jan Audestad**[b]

[a]*Ericsson AS, Olav Brunborgsvei 6, P.O Box 40, N-1361 Billingstad, Norway. Phone: +47 66 84 16 68. e-mail: etodvt@eto.ericsson.se*

[b]*Telenor AS, P.O. Box 6701 St.Olavs plass, N-0130 Oslo, Norway. Phone: 47 22 77 99 52. e-mail: Jan-Arild.Audestad@s.hk.telenor.no*

Abstract - As shown in [1], Access and Location transparencies defined for Open Distributed Processing (ODP) and TINA are unsufficient to support terminal mobility since interoperability between DPE platforms is not guaranteed at all times. In this paper, we study all the functions and computational objects necessary to support terminal mobility. Both forms of interactions, operational and stream-oriented, are considered. The solution proposed is generic and can be customised for both continuous and discrete mobility in either wireline or wireless networks.

BACKGROUND

The researched work described in this paper results from the mobility part of the TELEcom REsearch Programme TELEREP, carried out at the Center for Technology at Kjeller, UNIK [2]. The programme has been established to obtain practical experience with ODP/TINA methods and the implementation of TMN and IN functionality, and to study how security could be provided in the DPE environment. The programme started in 1995 and will run over several years. Initially the programme decided to base its work on a public domain version of CORBA [3], and will be re-evaluated at a later stage. One is currently studying both theoretically and practically, as part of a Ph.D. programme, how mobility and security can be provided transparently by the platform environment.

I. INTRODUCTION

There are defined two forms of interactions between computational objects: operational and stream-oriented [4]. Supporting terminal mobility consists therefore of enabling both forms of interactions between a Computational Object CO1 belonging to the terminal domain and another one CO2 belonging to the telecom system domain (See Figure 1)

II. ENABLING OPERATIONS BETWEEN THE TERMINAL DOMAIN AND THE TELECOM SYSTEM DOMAIN

Prior to any operational interaction (operation invocation), a binding must exist between the two objects. This is called implicit binding, i.e objects do not explicitly request establishment of the binding but the establishment is done by the DPE on the kernel transport network (kTN). This kTN is logically separated from the transport network supporting stream flows. Such a separation allows evolution of kernel communication independently of the evolution of the technologies used for stream flows transport [5] (see Figure 2).

It is observed that operational interactions between computational objects are only possible if the kernel transport network is operative and there is always connectivity between two arbitrary DPE nodes, i.e every DPE node is reachable from every other DPE node.

For a wired network the connectivity is ensured at configuration time, i.e necessary links between nodes are defined such that connectivity remains even if some nodes and links fails. The topology of the network is defined statically and will only be altered in case of a failure or reconfiguration. A DPE node can always determine how to reach another DPE node.

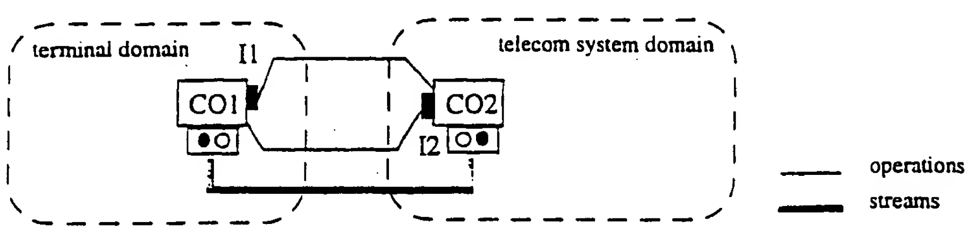


Fig.1 Interactions between COs residing the terminal and telecom system domains

This is not always true with terminal mobility. The mobile terminal is a particular DPE node with special behaviour:

• It changes frequently the node with which it has direct link.

• It may just disappear from a node and reappear later at any other node

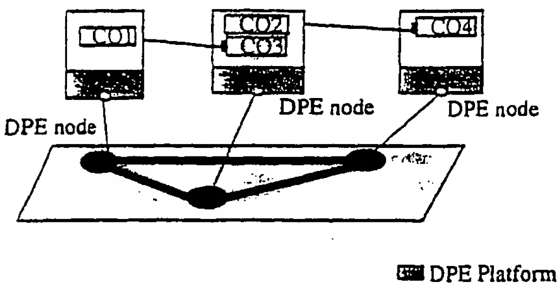


Fig.2 The kernel Transport Network

The topology of the kernel transport network for systems containing mobile DPE nodes is changing dynamically and more seriously, it can also be in an undetermined state in the sense that the connectivity with such a mobile DPE node is not always ensured unless additional functionally is inserted in the DPE.

We propose to consider the kTN as consisting of two parts:

• The fixed part comprising all fixed DPE nodes.

• The mobile part comprising all mobile DPE nodes.

At the boundary of the fixed part of the kTN there are several Network Access Points (NAP), i.e points where mobile DPE nodes (terminals) can link themselves to the fixed kTN (see Figure 3). It is worth noting that the NAP notion is different from the Network Termination Point (NTP) which designates the access point to the Transport network used for stream flows. As specified before the kTN and the Transport network are logically separate networks.

An **NAP** object is introduced to represent an NAP on the kTN. An **NAP** object is an interceptor which stands at the boundary between the terminal domain and the telecom system domain and is responsible for checking, transforming and forwarding of interactions that cross the boundary. An **NAP** object has two communication interfaces, one with the mobile DPE and one with the fixed kTN. Several **NAP**s can be located at the same DPE node.

Each mobile DPE node may have one or several Terminal Access Points (TAP), i.e. the points where the mobile DPE node can exchange operations with other DPE nodes. Here again, it is important to differentiate between a TAP and a Terminal Termination Point (TTP) which is used for the connection of stream flows. A TAP will be represented by a **TAP** object in the terminal domain.

Prior to any operational interaction the mobile DPE, i.e one of its TAPs, must be attached to an NAP. Operational interactions between a computational object residing on a mobile DPE and a computational object residing on a fixed DPE always go through a TAP and an NAP.

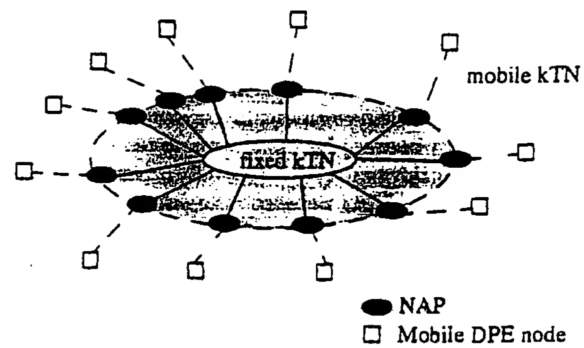


Fig.3 The kTN consisting of a fixed and a mobile part

A. Interactions initiated by the fixed part

When `CO2` invokes an operation `opY` on `CO1` (see Figure 4), the operation must be sent via the correct **NAP** object, i.e the one that is currently connected with the **TAP** of the mobile DPE. Since the terminal is moving, the **NAP** to which the **TAP** is connected may change from time to time.

To unburden `CO2` with mobility-related functions, a **`Terminal_Agent`** Object (**TA**) is introduced to undertake a kind of relocation function as shown in Figure 4. It keeps track of which is the current **NAP**. This relocation function is somewhat different from a relocation function defined in ODP. The latter records the change in location of an object when its is moved from one computing node to another. But in fact, recording the current **NAP** is semantically the same as recording the location of the terminal and hence the location of object `CO1` because the location of a terminal can be deduced from the location of the **NAP**.

For each mobile DPE node (or terminal) one instance of the **`Terminal_Agent`** object will be instantiated.

Instead of issuing an operation request directly to `CO1`, `CO2` issues a request to the **`Terminal_Agent`**. The **`Terminal_Agent`** will then forward it to the appropriate **NAP**. The **NAP** transfers it to the **TAP** which finally delivers it to `CO1`. To support interactions initiated by the fixed part, the **`Terminal_Agent`** object is therefore required.

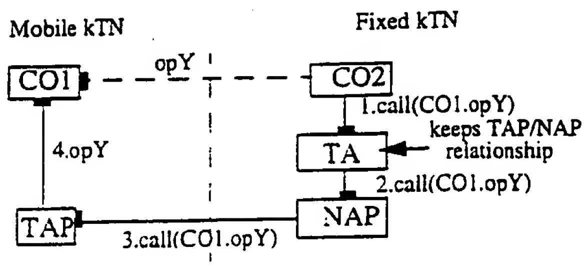


Fig.4 Interactions initiated by the fixed part

B. Interactions initiated by the mobile part

Suppose now that **CO1** wants to invoke an operation opX on **CO2** (Figure 5). First, the invocation must be conveyed to a **TAP**. This can be easily done assuming location transparency locally on the mobile DPE node. The **TAP** must then know to which **NAP** the operation should be forwarded. From the **NAP** the invocation can be conveyed to the **TA** (**Terminal_Agent**) and then to **CO2**, using access and location transparencies in the fixed kTN.

It is crucial that the **TAP** knows which **NAP** instance to communicate with in order to send the invocation to it. Since the **TAP** resides in a mobile DPE, the current **NAP** may be changing from time to time. The terminal mobility management consists in keeping track of the correct **NAP** instance. This responsibility may be allocated to the **SPA** object (Service Provider Agent).

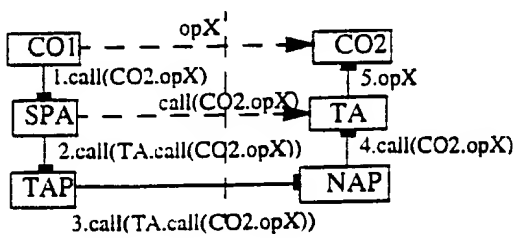


Fig.5 Interactions initiated by the mobile part

The **SPA** is thus entrusted with two responsibilities: supporting security functions and keeping location information. In this way, only one interceptor object is required in the terminal for managing both security and location updating. The introduction of the **SPA** is also convenient to keep the **TAP** hidden from the application objects. Instead of issuing an operation invocation to **CO2**, **CO1** issues a request to the **SPA**. The reason is that a mobile DPE, for example a PABX may in fact have several **TAP**s which should be transparent to the application objects. The **SPA** will ensure this transparency. Interactions initiated from the mobile DPE are realised as shown in Figure 5:

We may let the **SPA** involve in interactions initiated by the fixed part in order to obtain a symmetric interceptor configuration. Combining both types of interactions, the conveyance of operations across the boundary between the mobile DPE and the fixed DPE will be as shown in Figure 6.

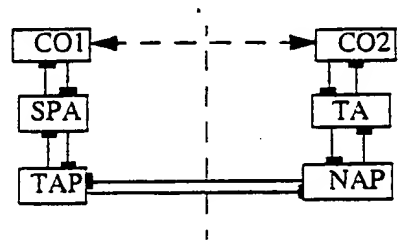


Fig.6 The conveyance of operations between the mobile DPE and the fixed DPE

C. Location registration and deregistration

It is observed that the conditions for success of both types of interactions is the definition of the association between the **TAP** and the **NAP** and the definition of the association between the **TA** and the **NAP**. If one or both of the two associations are undefined, the interaction will be unsuccessful. Second, the two associations must be consistent with each other. If the **TAP** is associated with an instance of **NAP**, then the **TA** must be associated with the same instance of **NAP**. An inconsistency will lead to failure.

When the mobile DPE is moving, the **TAP** - **NAP** association and the **NAP** - **TA** association must change simultaneously and consistently, and may sometimes be undefined. The operations necessary to determine these two associations are commonly referred to as **location registration** and **location deregistration** [6]. The various methods used to determine these two associations constitute therefore the different strategies for location registration and location deregistration. Below we shall look at several such methods and identify how they can be supported by the objects proposed above. Some of the methods will require additional objects. We will, however, show that the basic structure proposed above will be retained also in all cases we have identified. However, the different methods will require that the **TAP**, **NAP** and **TA** objects are equipped with the appropriate algorithms depending on the method adopted for a given system.

0.1 The wireline case

In the wireline case, the association between the **TAP** and the **NAP** is directly reflected by and is equivalent to the physical link between the mobile DPE and the telecom system domain. The association between the **TAP** and the **NAP** is also equivalent to the association between the **TA** and the **NAP**. We have:

Physical link <=> Association TAP - NAP <=> Association TA - NAP

A change in state of the physical link results in the change of the association between the **TAP** and the **NAP** and consequently a change of the association between the **TA** and the **NAP**. The wireline terminal can be in one of two states: *registered* when the physical link is up and the association between the **TAP** and the **NAP** and the association between the **TA** and the **NAP** are defined; *deregistered* when the physical link is down and both associations are undefined. There is no intermediate state.

Terminal state	Physical link	Ass.TAP-NAP $\wedge$ Ass. TA-NAP
registered	Up	Defined
deregistered	Down	Undefined

Fig.7 States of the wireline terminal

The "physical link surveillance" method

A change of the physical link can be used to trigger both a transition in terminal state and a location registration or deregistration. The transition of the terminal state is shown in figure below:

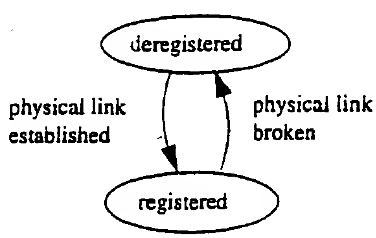


Fig.8 The state transition diagram of a wireline terminal

When the physical link is broken, i.e the terminal is unplugged from the socket or switched off or the line is broken, the association between the **TAP** and the **NAP** is removed. The **NAP** will immediately discover the situation and notify the **TA**. The **TA** set its **NAP** pointer to Nil. From now on, any attempts to invoke operations on the mobile terminal will be denied by the **TA**. The terminal is in the *deregistered* state.

When the physical link is established, i.e the terminal is plugged into the socket or switched on, the association between the **TAP** and the **NAP** is defined. The **NAP** will immediately discover the situation and notify the **TA**. The **TA** set its **NAP** pointer to the corresponding **NAP**. Interactions are then enabled. The terminal is in the *registered* state.

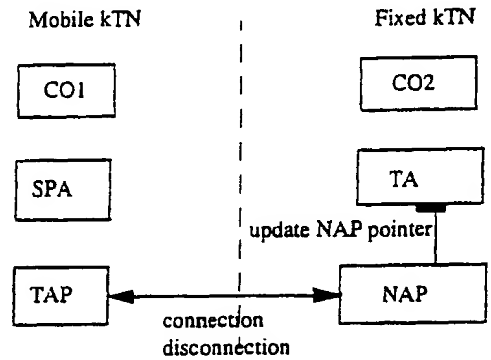


Fig.9 Location updating for wireline terminal

D. The wireless case

The wireless terminal stays most of the time in the disconnected state. When the physical link is down, it does not necessarily mean that the association between the **TAP** and the **NAP** is removed. The mobile DPE may still be there and, upon request, the physical link can be re-established. The mobile DPE may also have moved out the area. Hence, the state change of the physical link does no longer reflect the state change of the association between the **TAP** and the **NAP**.

Other methods to detect and update the change of the association between the **TAP** and the **NAP** are required. There are two alternatives: the initiative is taken either by the **NAP** or by the mobile DPE.

1) Initiative by the NAP

The NAP's initiative method

The detection of the changes of the association between the **TAP** and the **NAP** means that the **NAP** is able to find out which mobile DPEs are located in its area. The **NAP** must therefore have the capability to store the identifier of the **TAPs** which have been previously connected to it, i.e before the physical link is "broken".

Periodically, the NAP broadcasts a "hello" in its area as shown in Figure 10. Any mobile DPE present in the NAP's area will respond to the broadcast with its TAP identifier. The NAP will then compare the received TAP identifiers with the stored ones. For mobile DPEs previously registered, there is no change in the state of the association between the TAP and the NAP and no action is necessary. For mobile DPEs which have just arrived, the NAP issues an update call to the corresponding TAs. The TAs will set their NAP pointer to the NAP. For mobile DPEs which have left, the NAP also initiates an update of the corresponding TAs indicating that the mobile DPE did not respond. The TAs will reset their NAP pointer to Nil.

If the intervals between consecutive detection processes are short enough, the representation of the association between the TAP and the NAP in the telecom system domain can be considered to be in agreement with the real association. The association is also determined for all the terminals, i.e the telecom system domain has the status of all the terminals. A terminal is registered when the TAP and the NAP association is defined and deregistered when the association is undefined. In registered state all interactions from and to the mobile DPE are possible. In deregistered state no interaction is possible.

This solution can be used for systems where the status of the terminals are important. It can be used in systems with larger number of terminals but less geographically distributed, i.e small number of NAPs. For systems with large number of NAPs, this solution has many disadvantages. First, the NAPs must have storage capacity and consequently must keep the stored data, i.e TAP identifiers of mobile DPEs present in its area, up-to-date and consistent. Second, there are periodically much processing activity in every NAP. The third disadvantage is the inefficient use of the radio frequency or infrared access channel to the NAP which is a scarce resource shared by all mobile DPEs for both data transmission and signalling. When all the present mobile DPEs answer, much capacity is used just for detection of the DPEs.

2) Initiative by the mobile DPE

a. The Periodic method

The detection and updating of the change of the association between the TAP and the NAP can also be initiated by the mobile DPE. Each mobile DPE can periodically report itself to the nearest NAP. The NAP will then send a register request to the corresponding TAs. In order to detect the silent terminals, i.e those that have disappeared without a deregistration, each TA can be equipped with a timeout. If a mobile DPE does not register itself within a period of time t_0, its TA will set it as deregistered. By this method, the status of all the terminals are always known by the telecom system domain. This method is suitable for systems with small number of terminals but which are more geographically distributed (larger number of NAPs).

It generates, however, activity both on the access channel and in the telecom system domain.

b. The method based on location changes

It is observed that the changes in the association between the TAP and the NAP are caused by the mobility of the terminal. If the mobile DPE knows that it has moved from one NAP to another, it also knows that the association between the TAP and the NAP has to be updated. For this to work the mobile DPE need to have the capacity to store the identifier of the previous NAP and the capability to obtain the identifier of the current NAP.

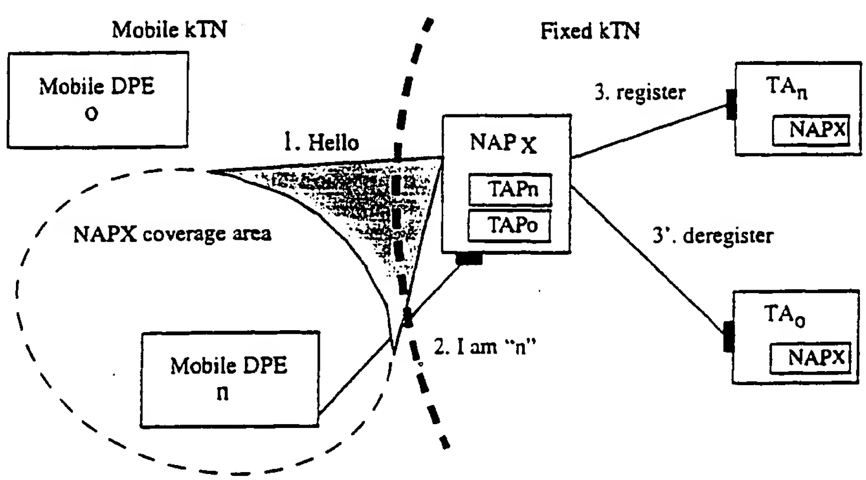


Fig.10 The case of detection function assumed by the NAP

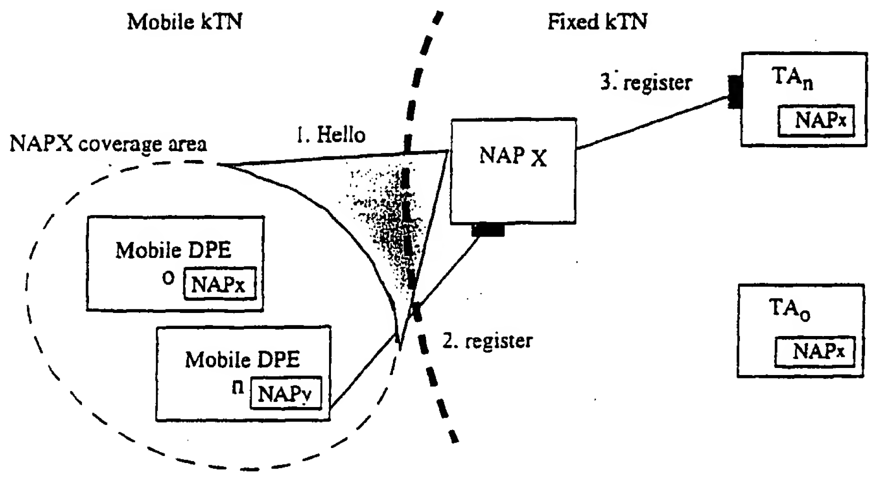


Fig.11 The case of detection function assumed by the mobile DPE

As shown in Figure 11. an **NAP** is associated with a geographic area called NAP coverage area. The **NAP** broadcasts periodically its identifier in this coverage area. Every mobile station in the coverage area reads periodically this **NAP** identifier and compares it with the one stored previously. If they are similar. there is no change in the association between the **TAP** and the **NAP** and no updating is necessary. If they are different, there is a change in the association. The mobile DPE updates its local **NAP** pointer and requests a `register` operation on its **TA**. The request is sent to the **NAP** which forwards it to the corresponding **TA**. The **TA** sets its **NAP** pointer to the current **NAP**.

This alternative requires that the mobile DPE has some storage capacity and some intelligence to decide whether a registration is necessary or not. On the other hand, the **NAP** does not have to store and process the information about the mobile DPEs present in its area. The use of the access channel to detect and update a change in the association between the **TAP** and the **NAP** is more optimal, since the probability that all the mobile DPEs will change **NAP** is very small and hence a total registration of all DPEs will be very infrequent.

The solution used in GSM [7] is slightly different from the one presented above. A base station can be mapped to an **NAP** in our model. There is however an intermediary object between the **NAP** and the **TA**. As shown in Figure 12 this object can be modelled as a group of **NAPs**, **GNAP** or as a "mirror" **TA**, **MTA**, holding some data for the **TA**. The location updating is executed only when the mobile station has moved from one **GNAP** or **MTA** to another. When required, e.g for initiating a call to the mobile DPE, the determination of the **NAP** is done "on the fly" by the **GNAP** or **MTA**.

There is, however, a problem. The updating of the associations between the **TAP** and the **NAP** and the association between the **TA** and the **NAP** are initiated by the mobile DPE when it knows that a change has occurred. It is not always true that the mobile DPE is aware of such a change and capable of notifying the telecom system domain. It can be switched off and brought to another area, a fault may occur in the mobile DPE, or it may have run out of battery power. It can also move out of the coverage area, loose all contact with the telecom system domain and be unable to initiate the updating.

In such a situation there may or may not be a mismatch between the real association between the **TAP** and the **NAP** and the perception of the telecom system domain, i.e the state of the association stored in the telecom system domain. When a terminal is registered, i.e the associations between the **TA** and the **NAP** and the association between the **TAP** and the **NAP** are defined, it is not guaranteed that interactions from and to the mobile DPE are possible. Only when the physical link is in operation, i.e there is activities to or from the terminal, is the state well-defined. When the physical link is down, nothing more is known about the location of the mobile DPE than its previous registration.

In the telecom system domain it is useful to differentiate between the two registered states of the terminal, namely **registered and confirmed** state and **registered and unconfirmed** state. In addition to the **NAP** pointer the **TA** needs also to save the terminal state. On the mobile DPE side there is no need for additional state but the registered state and the deregistered state. The **NAP** pointer is sufficient to represent these two states.

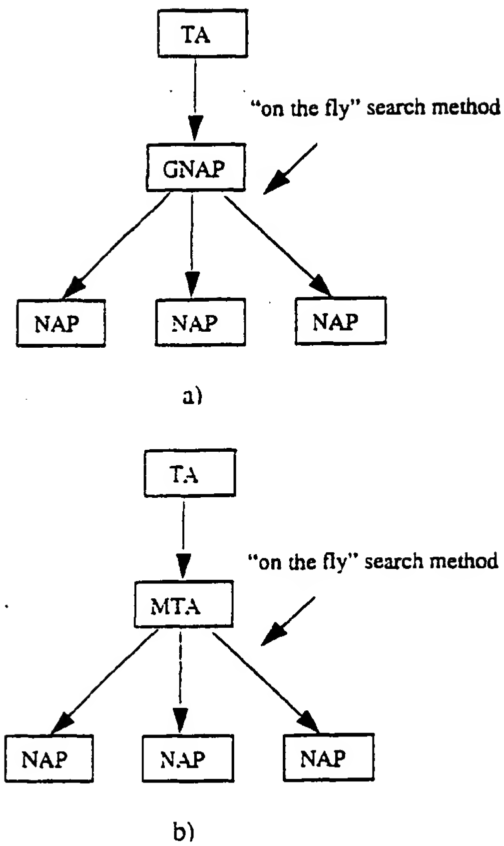


Fig. 12 Alternative models of GSM

Seen from the telecom system domain, a wireless terminal may therefore be in one of the following three states:

- **deregistered**: the mobile station has actively notified that it has been taken temporarily out of service.
- **registered and unconfirmed**: the mobile station is registered at an NAP but there is no activity on it, i.e the physical link is down.
- **registered and confirmed**: the mobile station is registered at an NAP and there is activity going on, i.e the physical link is up.

The states of the wireless terminal is thus defined by the state of the physical link, the association between the **TAP** and the **NAP**, and the association between the **TA** and the **NAP** as follows:

Terminal state	Ass.TAP - NAP ∧ Ass. TA - NAP	Physical link
deregistered	Undefined	Down
registered & unconfirmed	Defined	Down
registered & confirmed	Defined	Up

Fig. 13 States of the wireless terminal

From the registered and confirmed state the mobile DPE changes its state to the registered and unconfirmed state when all the activities have ceased. It will remain in that state until some successful activity takes place between the mobile DPE and the fixed DPE. The terminal state is then reset to the registered and confirmed state. If the request is unsuccessful, the terminal state will be changed to the deregistered state and will remain there until a registration takes place. From the registered and confirmed state, the terminal state can be changed to the deregistered state when the mobile DPE has explicitly made a deregistration request or has disappeared, i.e does not respond to an invocation initiated by the fixed DPE.

The transitions between the terminal states are shown in Figure 14. It is worth noting that in addition to normal transitions such as registration, activation, etc. there are two special transitions shown in Figure 14: handover and location updating. In fact, the handover and location updating can be regarded as a transition because the associations **TAP - NAP** and **TA - NAP** do change, but remain defined. Location updating may be regarded as a transition from the registered and unconfirmed state back to the registered and unconfirmed state. Similarly, handover may be regarded as a transition from the registered and confirmed state to the registered and confirmed state. A handover may follow a location updating or may be executed alone. From one registered and confirmed state, the terminal may also migrate to another registered and confirmed state after a location updating alone without a handover.

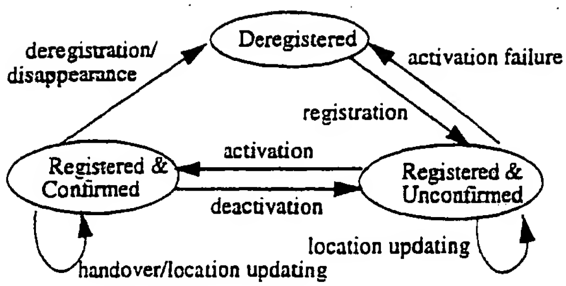


Fig.14 The state transition diagram of the wireless terminal

III. ENABLING STREAM BINDING WITH THE MOBILE TERMINAL

We assume now that operational interactions can be established between the mobile terminal and the telecom system domain. From now on, in order to reduce the complexity and make the explanations easier, operations across the boundary between the mobile DPE and the fixed DPE will be denoted as if they are directly addressed between peer objects, although they still have to go through the objects **TA**, **NAP**, **TAP** and **SPA**.

The transport network used for stream flow can be modelled as network delimited by two or more NTPs (Network Termination Points) and capable of transporting bit stream between two NTPs (see Figure 15). An NTP is the point at which a bit stream is accepted and/or delivered. Associated with an NTP is the characteristic of the stream accepted/delivered at the NTP, such as frame structure, QoS, etc. Different NTPs may have different characteristics.

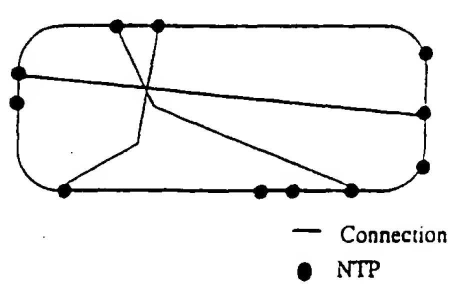


Fig.15 The transport network for stream flow

TINA has defined a similar model called the connectivity layer network [8]. However, the term Access Point is used instead of NTP. There is actually no difference between these two terms. The reason that the term NTP is used in this paper is just to differentiate from the NAP (Network Access Point) which is the access point of the kTN (kernel Transport Network). It is very important to make a distinction between the kTN and the transport network for stream. As mentioned in [5], these networks may be physically separate or share the same physical resources and represent only different concepts of the network. By separating the two network concepts, more freedom of choice for the kernel transport network is obtained and the evolution of the kernel transport network is made independent of the evolution of the technologies used for stream flow transport. The NTPs can be grouped into an NTPPool for example according to their geographical location. The area covered by the telecom system domain can hence be divided into smaller areas, each one served by one NTPPool.

As shown in Figure 16, when the mobile terminal is in area A, a stream flow between an object residing in the terminal and another object residing in the telecom system domain goes through a given NTPa belonging to $NTPPool_a$. When the terminal leaves area A and enters, for example area B, then its stream flows must go through NTPb belonging to $NTPPool_b$. $\mathbf{TCSM_t}$ must somehow be aware that the terminal has moved and that another NTP belonging to another NTPPool should be used.

For each NTPPool let us define an object called **NTP_Manager**. The **NTP_Manager** is in charge of an area and has information about all NTPs in that area, i.e which NTPs are available and their characteristics (QoS parameters). The **NTP_Manager** is periodically broadcasting its identifier in its area. The **NTP_Manager** identifier will be used by the $\mathbf{TCSM_t}$ when it wants to communicate with the **NTP_Manager**.

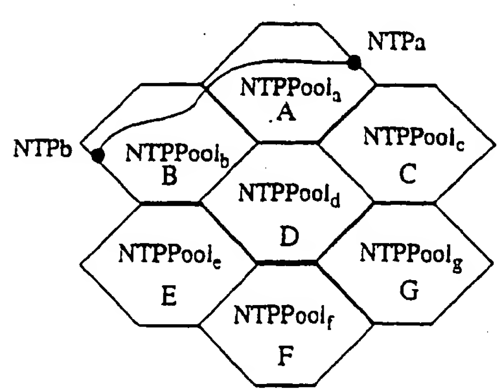


Fig.16 The coverage area divided into area served by NTPPools

When the terminal t is entering area A, the **TCSM$_t$** will get the identifier of the **NTP_Manager$_A$**, which is received through the **TAP**. Note that the kTN is used for the broadcast of NTP information. Upon a connection request from **CSM**. **TCSM$_t$** will ask the **NTP_Manager** to allocate an NTP for its TTP. The **TCSM$_t$** will return the identity of this NTP back to **CSM**. **CSM** can then request the **CC** to establish a connection between this NTP to be used for **CO$_t$** object and the NTP to be used for **CO$_n$**.

Actually, the **TCSM** of the fixed node closest to the area A may be in charge of the allocation of NTPs. However, it is better to place all the connectivity functions which are specially related to terminal mobility in a dedicate object such as the **NTP_Manager** which can be modified without any consequence for the **TCSM**. Furthermore, according to TINA [9], a **TCSM** is defined for a node in the transport network and the notion of coverage area presented here has nothing to do with a node. A coverage area may or may not correspond to a node. A node may in fact contain several areas when it has numerous NTPs covering a large geographical area which need to be grouped into smaller areas.

The area covered by the **NTP_Manager** may or may not correspond to the area covered by the **NAP** since the two objects may be responsible for the connectivity with two separate physical networks, the kernel transport network and the transport network supporting streams.

There are three correspondence alternatives:

1. an NAP's area corresponds to a **NTP_Manager**'s area.
2. an NAP's area corresponds to several **NTP_Manager**'s areas.
3. an **NTP_Manager**'s area corresponds to several **NAP**'s areas.

The choice of the appropriate correspondence alternative for a particular system is a matter of dimensioning and configuration and will not be studied further here.

A. Handover

When the mobile DPE moves from one NTP coverage area to another, there are two ways to support the availability of service:

- **Continuous terminal mobility** ensures the continuity of service by using the continuous handover mechanism. GSM is an example of this type of terminal mobility [7].
- **Discrete terminal mobility** ensures only the continuity of the service sessions but not the continuity of service, i.e the service sessions are suspended before and during the transition and resumed when the mobile DPE has reached the new NTP coverage area. This is referred as discrete handover or session mobility.

We shall now consider both cases.

B. Continuous handover

To ensure that stream flows are not disrupted when the mobile terminal is moving from one NTP coverage area to another, there must be an overlap between coverage areas as shown in Figure 17. In the intersection area $A \cap B$ of two NTP coverage areas A and B, the mobile terminal has contact with both **NTP_Managers** and stream flows can be established with NTPs belonging to both NTPPools.

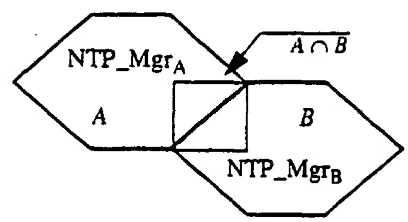


Fig.17 Overlap between NTP coverage areas

1) Handover procedure

A handover procedure consists of the following steps:

1. To detect that handover may be necessary. This can be done via the detection that the mobile DPE is in an intersection area such as $A \cap B$ or $A \cap B \cap C$, etc.
2. To decide that handover must be executed. The decision can be taken based on different criteria such as transmission quality, distance between the mobile DPE and the NTP, etc. The criteria are system dependent and cannot be implemented in a generic way. Furthermore, the decision for handover can be made by the mobile DPE or by the telecom system domain or also by both.
3. To perform the handover in the fastest and most secure way.
4. To move to a new stable state.

2) Necessary computational objects

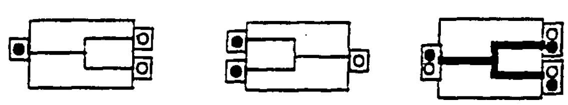
In order to realise the procedure described above, several new objects are required in the system.

First, an object called **Handover_Initiator** is introduced to decide and supervise the handover procedure. The **Handover_Initiator** can be created by the **TCSM** on

the mobile DPE if the handover decision should be made by the mobile terminal. It can also be created by the **NTP_Manager** on the fixed DPE if the handover decision should be made by the telecom system domain. The **Handover_Initiator** object encapsulates the functions necessary to realise a specific decision criterion. Hence, for a specific system using a specific criterion, a specific version of the **Handover_Initator** is needed. All the **Handover_Initiator** objects have, however, the same operational interface toward other objects in the system.

Second, two objects called **Handover_Source** and **Handover_Sink** are introduced. Each object has at least three stream interfaces. The **Handover_Source** object has the ability to connect an input stream interface with two output interfaces and to duplicate the information flow form the input stream interface. The **Handover_Sink** object has the ability to connect two input stream interfaces into one output stream interface and to discard the duplicated information from two identical input streams. In the case where there is a bidirectional stream flow between two objects, it is more convenient to use an object called **Handover** having the functionality of both the **Handover_Source** and **Handover_Sink** as shown in Figure 18. The binding of stream interfaces belonging to the same object is described in [10].

The **Handover_Source** should be inserted after the source object using the stream direction as reference and the **Handover_Sink** object should be inserted before the sink object.



a. Handover_Source   b. Handover_sink   c. Handover

Fig.18 The Handover objects

It is worth noting that the **Handover** objects in fact represent a combination of both hardware and software modules which are specific to a system. Although these objects may have different internal functions and mechanisms they will have the same external interfaces.

3) Example of handover

Suppose that the mobile terminal is originally in area A and here is a stream flow between its object **CO**$_t$ and another object **CO**$_n$ in the telecom system domain. The stream goes hrough TTP_a, NTP_{AX} and NTP_u.We introduce now two objects **Handover**$_t$ having three stream interfaces St1, St2, St3 and **Handover**$_n$ having three stream interfaces Sn1, Sn2 and Sn3. We let the mobile DPE make the handover decision through a **Handover_Initiator** object residing in the mobile DPE.

a. Before handover

When the **CSM** receives the request to bind **CO**$_t$ and **CO**$_n$, it knows that **CO**$_t$ is residing on a mobile DPE by reading the name of **CO**$_t$. It may then deduce that a handover object is necessary. It creates or requests a Service Factory object to create an object **Handover**$_n$ and proceeds with the procedure to establish a stream between **CO**$_n$ and the stream interface Sn1 of **Handover**$_n$. This is done without problems since both **CO**$_t$ and **Handover**$_n$ are residing in the fixed kTN. **CSM** requests **Handover**$_n$ to bind internally the stream interfaces Sn1 and Sn2 as shown in Figure 19

Towards the mobile side, **CSM** requests **TCSM**$_t$ to establish a local connection for **CO**$_t$. **TCSM**$_t$ knows that it is residing on a mobile DPE and that a handover object is necessary. It creates or requests a Service Factory object to create an object **Handover**$_t$. It will then establish a stream between **CO**$_t$ and stream interface St1 of **Handover**$_t$. **TCSM**$_t$ will then ask **Handover**$_t$ to bind internally the stream interfaces St1 and St2. **TCSM**$_t$ requests also **Handover**$_t$ to bind St2 to TTP_a. **TCSM**$_t$ returns the corresponding NTP identifier to which TTP_a is connected, namely NTP_{AX}, to the **CSM**.

The **CSM** then connects the **Handover**$_n$ object to NTP_{AX}. When this is accomplished, a stream is established between **CO**$_t$ and **CO**$_n$.

When the terminal moves into the intersection area $A \cap B$, **TCSM**$_t$ will receive the identifier of the **NTP_Mgr**$_B$ and know that a handover may be necessary. It creates or requests a Service Factory to create a **Handover_Initiator** object in the mobile DPE.

The **Handover_Initiator** encapsulates the mechanism for handover decision and starts to operate immediately after its creation.

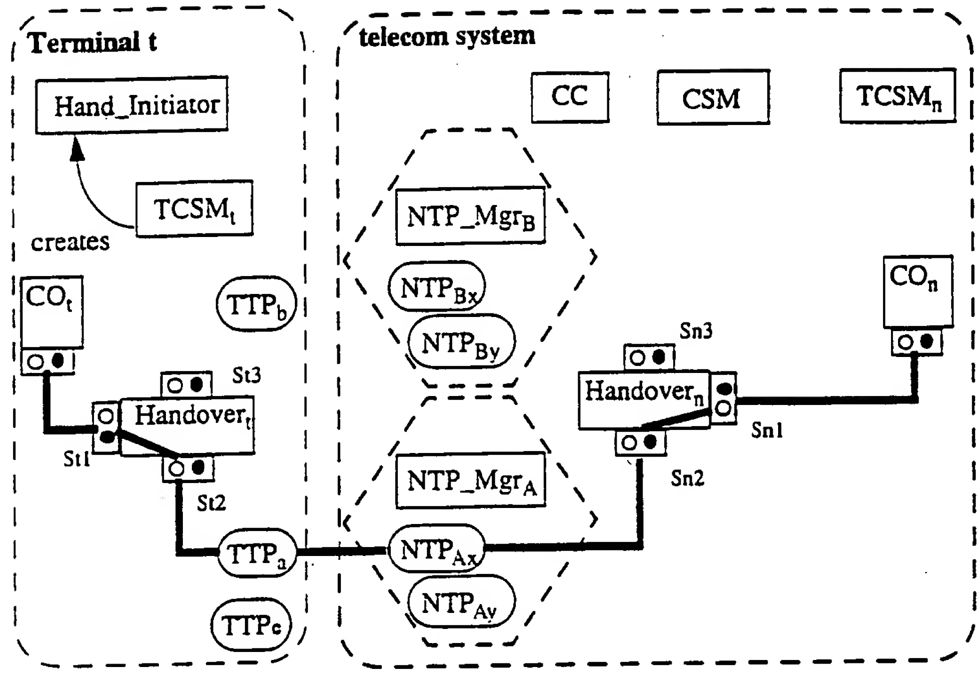


Fig.19 The stream flow before handover

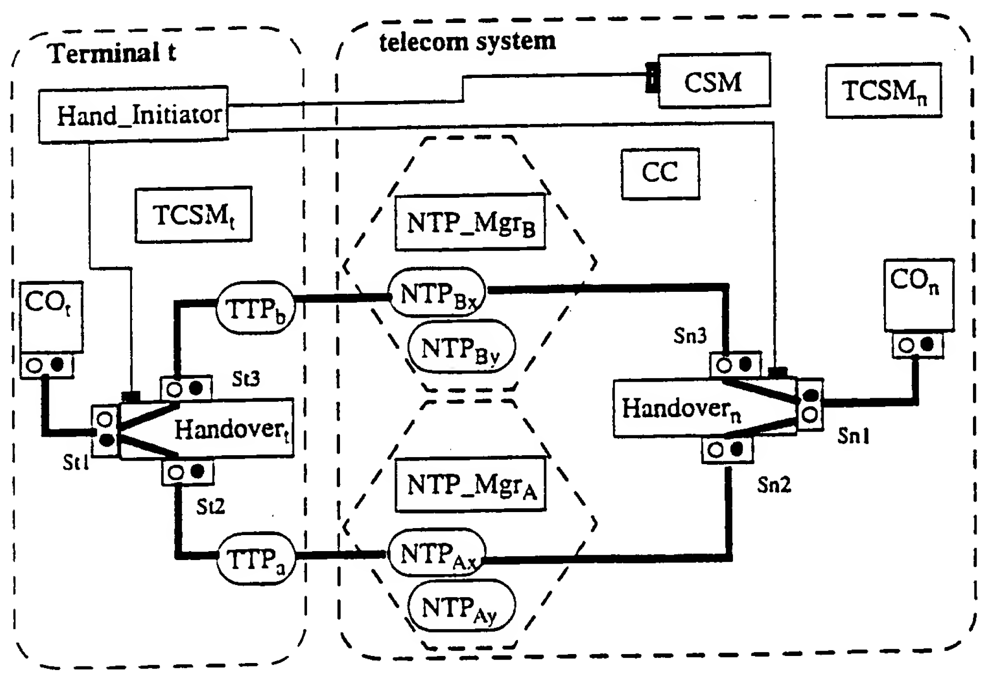


Fig.20 The stream flow during handover

b. During handover

When the criterion for handover is fulfilled, the **Handover_Initiator** will start the handover procedure. It requests **CSM** to establish a stream between the interface St3 of **Handover$_t$** and the interface Sn3 of **Handover$_n$**. **CSM** requests the **TCSM$_t$** to bind St3 to a TTP. The **TCSM$_t$** binds St3 to TTP_b. **TCSM$_t$** returns the corresponding NTP identifier to which TTP_b is connected. namely NTP_{BX}, to the **CSM**. **CSM** then connects the **Handover$_n$** object to NTP_{BX}.

Everything has now been prepared for the switchover. The **Handover_Initiator** requests **Handover$_t$** to bind internally the interfaces St1 and St3, and **Handover$_n$** to bind internally the interfaces Sn1 and Sn3. The **Handover_Initiator** then requests **CSM** to release the stream between the interfaces St2 of **Handover$_t$** object and Sn2 of **Handover$_n$** object. It also requests **Handover$_t$** object to unbind internally St1 and St2 and **Handover$_n$** to unbind Sn1 and Sn2.

In Figure 20 we just show that there are interactions between the **Handover_Initiator** and the **CSM**, and **Handover$_t$** and **Handover$_n$** without any further specification in order to preserve clarity.

c. After Handover

As shown in Figure 21, after the accomplishment of the handover procedure the **Handover_Initiator** object will terminate itself. The stream between the objects **CO$_t$** and **CO$_n$** follows now the new path without disruption.

D. Discrete handover

Without overlap between the NTP coverage areas, the mobile DPE will loose all contact with the telecom system domain when it moves out of an NTP coverage area. The continuity of service cannot be ensured. However, the service sessions in use can be suspended in order to be resumed later when the mobile DPE arrives at the new NTP coverage area. In a way, the service sessions can be considered as non-disrupted. The continuity of sessions requires the intervention of the user, i.e the user must explicitly order the suspension of the sessions in use before moving to another NTP area. He must also explicitly resume these sessions later on.

V. CONCLUSION

We have shown how terminal mobility can be supported in TINA architecture. To support mobility, the following mobility computational objects are required:

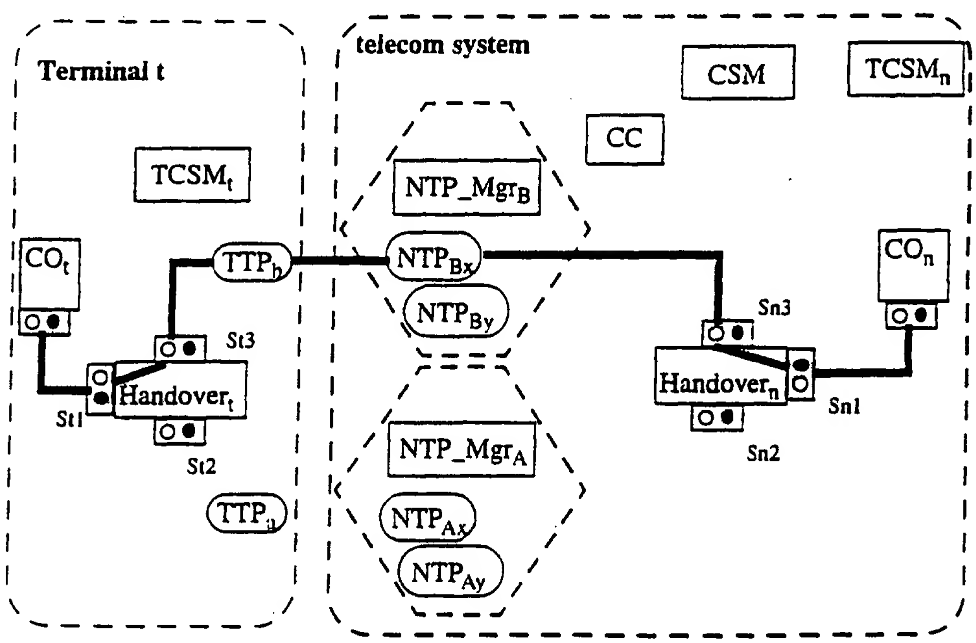


Fig.21 The stream flow after handover

a. To support operational interactions:

- In the mobile DPE:
  - **SPA**
  - **TAP**
- In the telecom system:
  - **TA**
  - **NAP**
  - **Terminal_Data**

b. To support stream flows:

- In the mobile DPE:
  - **Handover** objects
  - **Handover_Initiator**
- In the telecom system
  - **NTP_Manager**
  - **Handover** objects
  - **Handover_Initiator**

In addition, the following objects defined by the TINA Connection Management need to have enhanced functionality in order to cooperate with the mobility-related objects:

- **CSM**
- **TCSM**

These two objects must interact with the mobility objects such as the **TA**, the **SPA**, etc. in order to support streams. They must have the ability to distinguish between objects residing on a mobile DPE and objects residing on the fixed DPE, in order to decide about the creation of the **Handover** objects for streams spanning the terminal domain and the telecom system domain. The **TCSM** must also have the ability to detect that a handover may be necessary in order to create the **Handover_Initiator** object which controls the handover procedure.

REFERENCES

[1] v. T. Do, Jan A. Audestad. Mobility and TINA. In *Proceedings of the TINA 96 Conference*, volume ISBN 0. TINA-C, September 1996. Heidelberg, Germany.

[2] P. Spilling. The telecom research program telerep at unik. *UNIK Aarsrapport 1996*, 1996. http:www.unik.no/ paal/ telerep.html.

[3] Object Management Group, Inc. *The Common Object Request Broker: Architecture and Specification*. Document No. 91.12.1, Revision 1.1, ISBN 0-471-58792-3.

[4] TINA-C. *Computational Modelling Concepts*. January 1995.

[5] J. A. Audestad. Introduction to distributed processing in telecommunications systems. Norwegian University of Science and Technology, Departement of Computer Systems and Telematics. February 1996. Lecture notes - Course 45356 Communication in distributed systems.

[6] ETSI TC-SMG. *European digital cellular telecommunications system phase 2: Location registration procedures*. september 1994.

[7] ETSI TC-SMG. *European digital cellular telecommunications system phase 2: Mobile Application Part (MAP)*. july 1994.

[8] TINA-C. *Unification of Connection/session Graphs, Stream Interfaces and Channel Model*. April 1996. TR_PL.001_1.3_95.

[9] TINA-C. *Connection Management Architecture*. March 1995.

[10] v. T. Do, Jan A. Audestad. Experiences using corba to build telecommunications applications. In *Proceedings of the IFIP/ICCC International Conference on Information Network and Data Communication*, number ISBN 0 412 75750 8. Chapman & Hall, June 1996. Trondheim, Norway.

BIOGRAPHY

Dr. Do van Thanh received a M.Sc. degree in Electronical Engineering and Computer Science from the University of Trondheim, The Norwegian Institute of Technology, Norway in 1984 and Ph.D degree in Informatics from the University of Oslo, Norway in 1997. He joined Ericsson in 1991 after having worked seven years for Norsk Data, a minicomputer manufacturer in Oslo. He is currently a senior research scientist at Ericsson Applied Research Center in Norway and is in charge for Mobility and Distributed Computing research activity.

Jan A. Audestad is a senior advisor to Corporate Management of Telenor and part-time Professor in Distributed Systems at the Norwegian University of Science and Technology. He has thirty year research experiences in telecommunications, covering Satellite Systems, land Mobile Systems (GSM), Intelligent Networks, Security and Telecommunications Policy. He has chaired a number of standardisation and research groups in ITU, ETSI and Eurescom.

Mobility and TINA Architecture

Do van Thanh [a] *and Jan A. Audestad* [b]

[a]*Ericsson AS c/o Center for Technology at Kjeller, P.O Box 70, N-2007 Kjeller, Norway. Phone: 47 63 814570, e-mail: dvt@unik.no*
[b]*Telenor AS, P.O. Box 6701 St.Olavs plass, N-0130 Oslo, Norway. Phone: 47 22 779952 e-mail: Jan-Arild.Audestad@s.hk.telenor.no*

1 Introduction

The main goal of the SANDRA (Services And Network Database Research Activity) project at the Center for Technology at Kjeller is to try out the concepts and principles of TINA [9] in the specification, design and implementation of telecommunications applications, especially mobile applications. One objective of the project is to make the realization of mobile applications simpler by masking from the application all the mechanisms necessary to handle mobility, i.e making mobility part of the engineering viewpoint. This is done partly by using transparencies defined by TINA and partly by identifying additionnal functions not yet supported by the TINA DPE (Distributed Processing Environment) [8], [5]. As a case of study, Universal Personal Telecommunications (UPT) [1], [3] will be designed and implemented in a TINA-compliant manner. A CORBA-implementation [4], ORBeline, from PostModern Computing Technologies, Inc. is used as TINA-DPE.

2 Mobility is seamlessly supported in an ideal model

Let us start with an ideal situation where there exists a large DPE underneath all the computational objects and supporting access and location transparencies. In such ideal model, an object User A can move anywhere and still receive telecommunication services. To reach User A, any other objects just need its reference or ID and uses access and location transparencies to find it. The same reasoning can be applied to a terminal. Thus, the access and location transparencies will ensure that a stream can always be established between two Terminal objects independently of their locations. It is therefore possible to conclude that:

In an ideal model where a large DPE supporting access and location transparencies is assumed, personal mobility and terminal mobility is seamlessly supported.

3 Mobility is not automatically supported in a real model

For two reasons, this model is too ideal. First, if the User is a human being, there is no DPE in the end-user. A User must, therefore, in general be considered as a non-DPE node. Further, the User (viewed as an object) cannot belong to the same administrative and technological domain [7] [2] as the telecommunications system. In the domain "Telecom System", it is necessary to introduce an interceptor (as defined in [2]), called PD_User_Agent (Provider Domain User Agent), which represents the User and is responsible for protection and security functions. All the requests addressed to a User are then issued to the PD_User_Agent. Personal mobility depends on the PD_User_Agent's ability to locate the user. Without this ability, personal mobility is not supported. Similarly second, a Terminal should also be considered as a separate domain. In the domain "Telecom_System", a PD_Terminal_Agent represents the Terminal. The Agent will be responsible for security functions, i.e identification and authentication of the terminal. Terminal mobility depends on the PD_Terminal_agent's ability to locate the terminal. It is therefore possible to conclude that:

In a realistic model, where domain concept are taken into account, personal and terminal mobility are no longer automatically supported by ODP transparencies alone.

4 Offering mobility transparency to the applications

In order to support personal mobility, the PD_User_Agent must be equipped with the function to store and update user location information It is worth noting that the location information is subject to frequent change due to the mobility of the user. The PD_User_Agent must have an operational interface allowing the User to register his location.

Concerning terminal mobility, the mobile terminal is a particular DPE node with special behaviour:

- It changes access node frequently.

- It may just disappear for a while and re-appear at a diiferent access node.

The topology of the kernel Transport Network (kTN) is changing dynamically and, more seriously, it can also be in an undetermined state. The connectivity with such a mobile DPE node is not always ensured unless additional functions are inserted in the DPE. We propose to consider the kTN as consisting of two parts:

- A fixed part consisting of all fixed DPE nodes.
- A mobile part consisting of all mobile DPE nodes.

At the boundary of the fixed part of the kTN, there are several Network Access Points (NAP), i.e points where mobile DPE nodes (terminals) can connect themselves to the fixed kTN. On each mobile DPE node, a Mobility_Mgr object assumes the responsibility of connecting the mobile DPE to the fixed kTN. Interactions between a computational object residing on a mobile DPE and object on a fixed DPE always go through a NAP object and a Mobility_Mgr object.. Supporting terminal mobility means therefore to maintain dynamically the associations between Mobility_Mgr and NAP and between NAP and PD_Term_Agent.

In order to verify all the concepts and ideas concerning the support of mobility, a simulated system is designed and built. Horizontally, the system is structured into separate functional layer: Service Layer, Mobility Layer and Resource Layer. Vertically, the system is divided into two domains: Terminal Domain and Telecom Domain (fig. 4.1).

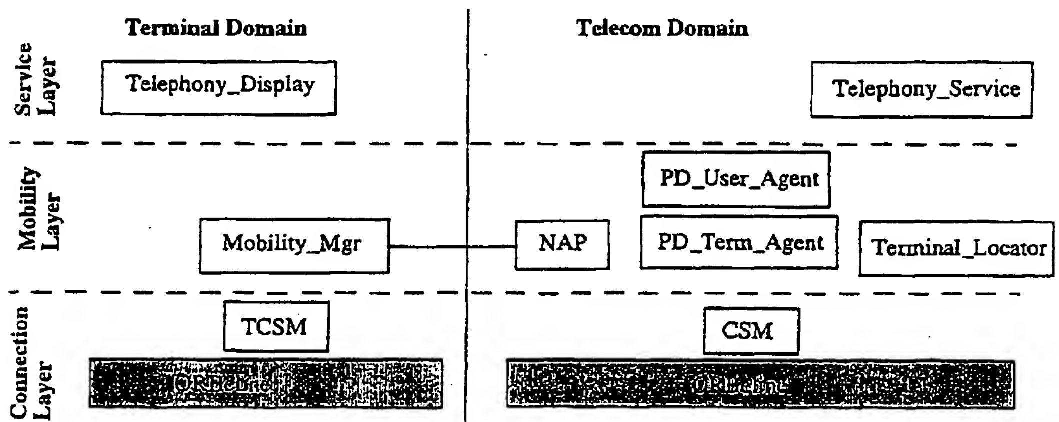


Figure 4.1: Simulated environement at the Center for Techonlogy at Kjeller

5 Conclusion

We have shown that personal and terminal mobility can be supported in TINA architecture. In order to offer mobility transparency to the applications, a functional layer called Mobility Layer is required. Such Mobility Layer facilitates the construction of mobile applications and promotes re-use in specification, design and coding. Our implementation is however rudimentary and several issues are left for further study such as replication and migation strategies for the agents, more advanced transaction mechanisms, etc.

More extensive paper on this problem can be obtained by request to Mr. Do van Thanh.

LITERATURE

[1] CCITT. *Principles of Universal Personal Telecommunication (UPT)*, 1992. Draft Recommendation F.850, version 4.

[2] ITU-TS. *Basic Reference Model of Open Distributed Processing - Part I Overview and guide to use the Reference Model.* Rec.X901(ISO/IEC 10746-1).

[3] ITU-TS. *Universal Personal Telecommunication (UPT) - Service Description*, April 1993. Draft Recommendation F.851, version 9.

[4] Object Management Group, Inc. *The Common Object Request Broker: Architecture and Specification.* Document No. 91.12.1, Revision 1.1, ISBN 0-471-58792-3.

[5] TINA-C. *Computational Modelling Concepts*, January 1995.

[6] TINA-C. *Connection Management Architecture*, March 1995.

[7] TINA-C. *Domain types and basic Reference Points in TINA*, May 1995.

[8] TINA-C. *Information Modelling Concepts*, April 1995.

[9] TINA-C. *Overall Concepts and Principles of TINA*, February 1995.